When Circuits Are Too Broad: Unit Tests for Mechanistic Interpretability

Anonymous Authors[1]

Abstract

Mechanistic interpretability claims often show that intervening on a circuit, feature, or representation changes a target behavior. Such target effects are necessary but insufficient: broad, lexical, confounded, or template-fragile interventions can produce the same evidence. We propose *mechanistic unit tests*, a negative-control protocol for evaluating the specificity of circuit and feature claims. The protocol asks whether a proposed mechanism survives nuisance rewrites, fails on matched negatives, avoids off-target damage, and dominates cheap same-budget baselines. We summarize these trade-offs with *specificity frontiers*, plotting target effect against off-target damage across intervention strengths. A controlled case study and a small `distilgpt2` pilot show how target-only evidence can hide lexical and negation failures. The contribution is a falsification layer for mechanistic claims, not a new discovery method.

1. Motivation

Many mechanistic interpretability methods can show that internal variables affect behavior: activation patching, causal tracing, mediation analysis, circuit discovery, sparse feature dictionaries, causal abstraction tests, and emerging benchmarks for causal localization and SAE evaluation (Vig et al., 2020; Meng et al., 2022a; Conmy et al., 2023; Bricken et al., 2023; Cunningham et al., 2023; Chan et al., 2022; Geiger et al., 2025; Mueller et al., 2025; Karvonen et al., 2025; Arora et al., 2024). What remains less standardized is the evidence that a proposed mechanism is *specific*. An intervention on variables H may change behavior Y while also damaging unrelated behaviors, exploiting a lexical artifact, or depending on the exact prompt template used to discover it. In safety-relevant settings, false specificity is especially dangerous: a circuit may appear to explain deception, refusal, or factual recall while actually tracking a prompt artifact or causing broad behavioral damage.

Software engineering often handles fragile claims about code by pairing tests that should pass with tests that should fail. We argue that mechanistic interpretability needs an analogous norm. A claim should routinely report not only "does this mechanism move the target?", but also "does it fail where the proposed explanation says it should fail?" This framing makes negative results informative: a failed unit test can identify where the explanation stops applying.

We contribute a method-agnostic reporting protocol for testing specificity. The key object is a *claim-test bundle*: a stated mechanism, intervention budget, invariance expectations, exclusion expectations, off-target probes, and same-budget baselines. This bundle adds a falsification layer to existing discovery methods, then summarizes the trade-off between target effect and collateral damage with a specificity frontier.

Our proposal is complementary to existing causal interpretability methods. Causal abstraction and causal scrubbing formalize whether a proposed high-level causal model is implemented by a network; activation patching and circuit discovery ask which components matter; MIB, CausalGym, and SAEBench compare methods and representations (Geiger et al., 2025; Chan et al., 2022; Zhang & Nanda, 2023; Heimersheim & Nanda, 2024; Mueller et al., 2025; Arora et al., 2024; Karvonen et al., 2025). Mechanistic unit tests ask a narrower reporting question: once a method proposes H, what contrast sets, rewrites, held-out controls, and same-budget baselines would make the claim fail? This scope lets the same reporting questions apply to circuits, SAE features, steering vectors, probes, model editing, and model repair (Meng et al., 2022a;b; Marks et al., 2025; Turner et al., 2023; Zou et al., 2023). The protocol also borrows deliberately from negative-control methodology and behavioral testing traditions (Lipsitch et al., 2010; Ribeiro et al., 2020; Gardner et al., 2020).

2. Mechanistic Unit Tests

Let a mechanistic claim have the form: internal variables H implement or mediate target behavior Y_t on distribution D_t. A unit test is a deliberately small perturbation of the claim

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

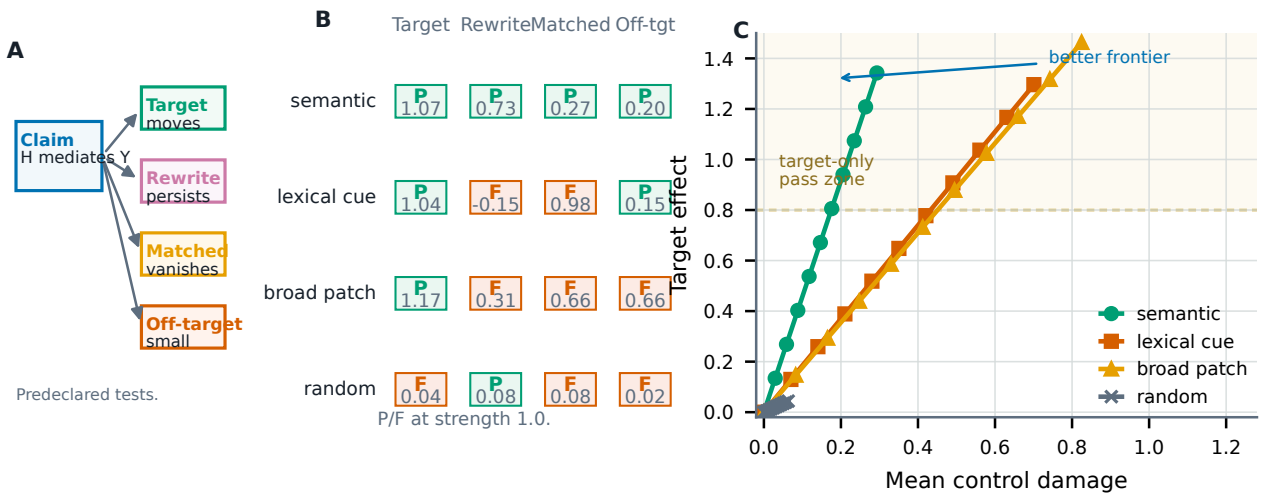

Figure 1. **Mechanistic unit tests turn target-effect claims into specificity claims.** A: each claim specifies expected passes and failures. B: the controlled case reports the full test vector at intervention strength 1.0; lexical and broad interventions pass the target-only check but fail controls. C: specificity frontiers sweep intervention strength. Up is more target effect, left is less control damage; the best frontier is therefore upper-left. The shaded band shows what target-only evaluation would accept while ignoring damage.

that should have a predictable outcome if the explanation is right. We propose four tests that can be attached to activation patching, feature ablation, steering, probing, circuit discovery, or sparse-autoencoder analyses.

Positive control. The proposed intervention should move the target behavior on the distribution used to state the claim. This is the usual causal-effect test and remains necessary.

Matched negative. The same intervention should not produce the target effect on examples that preserve superficial cues while removing the claimed mechanism, e.g., same entities but different relation. This test targets lexical, entity, and formatting confounds.

Nuisance rewrite. The effect should persist under meaning-preserving rewrites, e.g., paraphrases, tokenization variants, order changes, or language variants when the claim is semantic rather than template-specific. This test targets brittle prompt-template mechanisms.

Off-target probe. The intervention should avoid large changes on unrelated capabilities that the mechanism does not claim to mediate, e.g., syntax, factual recall outside the domain, arithmetic, refusals, or calibration. This test targets interventions whose apparent success comes from broader model damage.

3. Specificity Frontiers

Let a mechanistic hypothesis specify internal variables H, an intervention family I_λ with strength or size parameter λ, a target behavior Y_t, and a set of controls $\mathcal{C} = \{Y_1, \ldots, Y_m\}$. Define target effect

$$T(\lambda) = \mathbb{E}_{x \sim D_t}[\Delta(Y_t, M, I_\lambda(H), x)] \quad (1)$$

and off-target damage

$$D(\lambda) = \frac{1}{m} \sum_{j=1}^{m} \mathbb{E}_{x \sim D_j}[|\Delta(Y_j, M, I_\lambda(H), x)|]. \quad (2)$$

Here Δ can be a logit difference, probability change, accuracy change, calibration error, or task-specific scalar. Controls should be normalized by a predeclared tolerance, task range, or null-intervention variability before aggregation; papers should report both mean and maximum normalized damage so that a single catastrophic control failure is not hidden. The *specificity frontier* is the Pareto curve traced by $(D(\lambda), T(\lambda))$ across intervention strengths. A claim is stronger when its frontier reaches the same target effect with lower off-target damage, or higher target effect at the same damage, than same-budget baselines.

Baselines. The minimum comparison set is deliberately cheap: random same-size variable sets, high-activation variables, lexical-trigger variables, and same-layer patches of equal dimension. Baselines should match layer/site, dimensionality or sparsity budget, ablation type, corruption distribution, and metric choice, since activation-patching conclusions can be sensitive to these design choices (Zhang & Nanda, 2023; Heimersheim & Nanda, 2024; Syed et al., 2023). These baselines test whether the claimed mechanism improves over intervention power, activation magnitude, or prompt artifacts alone.

4. Protocol

The protocol is intentionally small enough to fit into existing mechanistic interpretability papers.

1. State a falsifiable claim: "variables H implement or mediate behavior Y_t under distribution D_t."
2. Specify the intervention family and expected sign of each test before seeing control outcomes.
3. Construct at least one matched negative, one nuisance rewrite set, and one off-target probe set.
4. Plot or tabulate the specificity frontier for the hypothesis and baselines.
5. Report failures: controls where the intervention moves too much, rewrites where the effect disappears, and baselines that match the claimed frontier.

A useful unit test has an expected direction, not just an expected magnitude. For example, if a claimed semantic direction is ablated, target confidence should fall on paraphrases that preserve the semantic fact; if the sign reverses under a superficial rewrite, the result suggests a boundary of the explanation. Thresholds should be chosen before evaluation and reported with enough raw effects for readers to apply stricter thresholds. At minimum, authors should report: target metric, control metrics, normalization denominator, target threshold, mean- and max-damage thresholds, bootstrap or seed intervals, number of controls, discovery/test split, and whether each control was predeclared or exploratory. Figure 1 summarizes the resulting reporting standard.

5. Controlled Case Study

We test the protocol in a controlled hidden-state system with rotated, mildly entangled latent variables: semantic content, a lexical surface cue, and an unrelated control feature. The target head uses both semantic content and a correlated lexical cue. Thus, ablating the lexical direction looks successful on the target distribution, even though the explanation "this is the semantic mechanism" is false. The controlled case study is not evidence that this failure mode is common in frontier models; its role is to isolate the logic of the test suite in a setting where the ground-truth mechanism and confound are known.

Figure 1 shows the expected behavior of the test suite in this controlled setting. A semantic direction changes the target and remains stable under meaning-preserving rewrites. The lexical cue nearly matches the target effect, but its sign flips under rewrite and it moves the matched lexical negative. The broad patch achieves an even larger target effect, but also causes high control damage. The random direction fails the positive control. Figure labels use predeclared illustrative thresholds: target effect at least 0.20, rewrite effect at least 0.50 times target effect, and each control damage below 0.40 times target effect; raw effects are reported so stricter thresholds can be applied. Across 50 seeds, the semantic direction has target effect 1.12 ± 0.03 and mean damage 0.24 ± 0.02, while the lexical cue has target effect 1.03 ± 0.03 but mean damage 0.56 ± 0.01; these are seed-to-seed sample standard deviations. Target effect alone would rank these mechanisms incorrectly; unit tests expose why.

6. Real-Model Pilot

To check that the protocol is operational in a standard mechanistic workflow, we run a small activation-intervention pilot on `distilgpt2`. We form final-token hidden-state directions at layer 4 from simple positive/negative review prompts, then ablate those directions at a predeclared strength $\lambda = 1.0$ while measuring next-token logit margins between `good` and `bad`. The target set uses held-out review prompts, rewrites replace the discovery verbs with paraphrases, matched negatives use negation while preserving surface cues, and off-target probes use unrelated factual next-token choices. The goal is not to establish a new sentiment circuit; it is to test whether target-only evidence survives the unit-test bundle.

Table 1. **Real-model pilot.** Both discovered directions move the target margin and survive rewrites, but fail matched negatives with negation. Matched and off-target columns report absolute control damage.

Candidate	λ	Target	Rewrite	Matched	Off-tgt	Verdict
sentiment	1.0	0.75	0.68	0.80	0.33	matched
love/hate	1.0	0.62	0.58	0.80	0.43	matched
random	1.0	-0.00	-0.00	0.01	0.01	low target

Table 1 shows the same pattern as the controlled case in an actual transformer: target-only evaluation would make both the sentiment and love/hate directions look promising, while matched negatives reveal that the claim scope is too broad. This is the intended use of mechanistic unit tests: they need not prove a mechanism correct; they make overbroad explanations fail visibly.

7. Reference Implementation

A small implementation can accept per-example target and control effects and return frontier points plus matched-grid comparisons. Such code is intended to sit downstream of activation patching in TransformerLens (Nanda & Bloom, 2022), nnsight-style tracing (Fiotto-Kaufman et al., 2024), sparse-autoencoder feature interventions, causal erasure/probing analyses, or activation steering. The controlled case study and figure are generated by scripts so that the

numeric summary and frontier plot can be regenerated from the same inputs.

Its purpose is to make the reporting standard easy to adopt and easy to criticize. In model case studies, the same fields shown in Figure 1 can be filled with actual target/control distributions and released with anonymized code when an artifact channel is available.

8. Positioning and Limitations

Unit tests are not a replacement for mechanistic understanding. A true mechanism may affect many behaviors because the model reuses a representation broadly. Conversely, a highly specific intervention may be brittle or artificial. Low off-target damage is therefore not a universal requirement for correctness. Damage should be interpreted relative to the scope of the claim: if a representation is claimed to mediate a broad capability, broad effects may be expected, but that scope should be stated before evaluation and tested against controls outside it. The proposed frontier is an evidential diagnostic, not proof that H is the model's native abstraction.

The protocol also depends on good controls. Poorly chosen negatives can make any method look specific, while overly broad off-target probes can punish genuine shared mechanisms. We recommend treating control design as part of the contribution and reporting the rationale for each control distribution.

The standard can also be gamed if controls are selected after seeing failures. For this reason, we recommend separating discovery examples from unit-test examples, reporting all attempted controls, and treating post-hoc controls as exploratory. This is the same discipline that makes negative results useful: a failed test should be visible even when it complicates the story.

9. Conclusion

Mechanistic claims are more useful when they specify how they could be wrong. Mechanistic unit tests make those failure conditions explicit through matched negatives, nuisance rewrites, off-target probes, and same-budget baselines. Specificity frontiers then show whether an intervention achieves target effects with low measured off-target damage. This standard does not prove a mechanism correct, but it makes overbroad explanations easier to detect and stronger explanations easier to compare.

References

Arora, A., Jurafsky, D., and Potts, C. CausalGym: Benchmarking causal interpretability methods on linguistic tasks, 2024.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N. L., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Hatfield-Dodds, Z., Tamkin, A., Nguyen, K., McLean, B., Burke, J. E., Hume, T., Carter, S., Henighan, T., and Olah, C. Towards monosemanticity: Decomposing language models with dictionary learning. Transformer Circuits Thread, 2023.

Chan, L., Garriga-Alonso, A., Goldowsky-Dill, N., Greenblatt, R., Nitishinskaya, V., Radhakrishnan, A., and Shlegeris, B. Causal scrubbing: A method for rigorously testing interpretability hypotheses. Alignment Forum, 2022.

Conmy, A., Mavor-Parker, A. N., Lynch, A., Heimersheim, S., and Garriga-Alonso, A. Towards automated circuit discovery for mechanistic interpretability. *Advances in Neural Information Processing Systems*, 36, 2023.

Cunningham, H., Ewart, A., Riggs, L., Huben, R., and Sharkey, L. Sparse autoencoders find highly interpretable features in language models, 2023.

Fiotto-Kaufman, J., Raghuram, V., Pal, K., Pu, X., Michaels, J., Kim, B., Chen, S., Huang, A., Zhu, C., Wei, Y., Biran, E., Singh, S., Geiger, A., Casper, S., Bau, D., and Marks, S. nnsight and ndif: Democratizing access to foundation model internals, 2024.

Gardner, M., Artzi, Y., Basmova, V., Berant, J., Bogin, B., Chen, S., Dasigi, P., Dua, D., Elazar, Y., Gottumukkala, A., Gupta, N., Hajishirzi, H., Ilharco, G., Khashabi, D., Lin, K., Liu, J., Liu, N. F., Mulcaire, P., Ning, Q., Singh, S., Smith, N. A., Subramanian, S., Tsarfaty, R., Wallace, E., Zhang, A., and Zhou, B. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP*, 2020.

Geiger, A., Ibeling, D., Zur, A., Chaudhary, M., Chauhan, S., Huang, J., Arora, A., Wu, Z., Goodman, N. D., Potts, C., and Icard, T. Causal abstraction: A theoretical foundation for mechanistic interpretability. *Journal of Machine Learning Research*, 26(83):1–64, 2025.

Heimersheim, S. and Nanda, N. How to use and interpret activation patching, 2024.

Karvonen, A., Rager, C., Lin, J., Tigges, C., Bloom, J., Chanin, D., Lau, Y.-T., Farrell, E., McDougall, C., Ayonrinde, K., Till, D., Wearden, M., Conmy, A., Marks, S., and Nanda, N. SAEBench: A comprehensive benchmark for sparse autoencoders in language model interpretability, 2025.

Lipsitch, M., Tchetgen Tchetgen, E., and Cohen, T. Negative controls: A tool for detecting confounding and bias in observational studies. *Epidemiology*, 21(3):383–388, 2010.

Marks, S., Rager, C., Michaud, E. J., Belinkov, Y., Bau, D., and Mueller, A. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models, 2025.

Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, 2022a.

Meng, K., Sharma, A. S., Andonian, A., Belinkov, Y., and Bau, D. Mass-editing memory in a transformer, 2022b.

Mueller, A., Geiger, A., Wiegreffe, S., Arad, D., Arcuschin, I., Belfki, A., Chan, Y. S., Fiotto-Kaufman, J., Haklay, T., Hanna, M., Huang, J., Gupta, R., Nikankin, Y., Orgad, H., Prakash, N., Reusch, A., Sankaranarayanan, A., Shao, S., Stolfo, A., Tutek, M., Zur, A., Bau, D., and Belinkov, Y. MIB: A mechanistic interpretability benchmark, 2025.

Nanda, N. and Bloom, J. Transformerlens, 2022.

Ribeiro, M. T., Wu, T., Guestrin, C., and Singh, S. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

Syed, A., Rager, C., and Conmy, A. Attribution patching outperforms automated circuit discovery, 2023.

Turner, A., Thiergart, L., Udell, D., Leech, G., Mini, U., and MacDiarmid, M. Activation addition: Steering language models without optimization, 2023.

Vig, J., Gehrmann, S., Belinkov, Y., Qian, S., Nevo, D., Singer, Y., and Shieber, S. Investigating gender bias in language models using causal mediation analysis. *Advances in Neural Information Processing Systems*, 33: 12388–12401, 2020.

Zhang, F. and Nanda, N. Towards best practices of activation patching in language models: Metrics and methods, 2023.

Zou, A., Phan, L., Chen, S., Campbell, J., Guo, P., Ren, R., Pan, A., Yin, X., Mazeika, M., Dombrowski, A.-K., Goel, S., Li, N., Byun, M. J., Wang, Z., Mallen, A., Basart, S., Koyejo, S., Song, D., Steinhardt, J., Hendrycks, D., Fredrikson, M., Kolter, Z., and Li, B. Representation engineering: A top-down approach to AI transparency, 2023.

A. Reporting Checklist

For each mechanistic claim, authors should disclose:

- **Claim scope:** behavior and distribution mediated by H.
- **Intervention budget:** layer, site, dimension, sparsity, and strength.
- **Invariance tests:** rewrites where the effect should persist.
- **Exclusion tests:** matched negatives where the effect should vanish.
- **Damage probes:** behaviors expected to remain stable.
- **Baselines:** same-budget random, high-activation, lexical, or site controls.
- **Predeclaration:** which controls were predeclared vs. exploratory.